



José Geraldo de Sousa Junior

**Uma Arquitetura para Aplicações Dinâmicas NCL
Baseadas em Famílias de Documentos**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial
para obtenção do título de Mestre pelo Programa
de Pós-Graduação em Informática da PUC-Rio.

Orientador: Prof. Luiz Fernando Gomes Soares

Rio de Janeiro
Novembro de 2011



José Geraldo de Sousa Junior

**Uma Arquitetura para Aplicações Dinâmicas NCL
Baseadas em Famílias de Documentos**

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Luiz Fernando Gomes Soares

Orientador
Departamento de Informática – PUC- Rio

Prof. Simone Diniz Junqueira Barbosa

Departamento de Informática – PUC-Rio

Prof. Carlos de Salles Soares Neto

UFMA

Prof. José Eugênio Leal

Coordenador Setorial do Centro
Técnico Científico – PUC-Rio

Rio de Janeiro, 09 de novembro de 2011

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

José Geraldo de Sousa Junior

Graduado em Ciência da Computação pela UFMA, onde participou de projetos na área da Engenharia Dirigida por Modelos. Integrou entre 2009 e 2011 o Laboratório TeleMídia da PUC-Rio, onde participou de projetos em aplicações hipermídia, mais especificamente na área de TV Digital.

Ficha Catalográfica

Sousa Junior, José Geraldo de
Uma arquitetura para aplicações dinâmicas NCL baseadas em famílias de documentos / José Geraldo de Sousa Junior ; orientador: Luiz Fernando Gomes Soares. – 2011.
72 f. : il. color. ; 30 cm
Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2011.
Inclui bibliografia
1. Informática – Teses. 2. NCL. 3. Televisão digital interativa. 4. Aplicações dinâmicas. 5. Templates. 6. Autoria orientada a templates. 7. Arquiteturas de software. I. Soares, Luiz Fernando Gomes. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

A Geraldo e Elbanice, pelo amor e apoio incondicionais.

A Thiago e Ariana, pelo companheirismo e força que recebo deles.

A Deus, por colocar pessoas especiais em minha vida.

Agradecimentos

Agradeço ao meu orientador Prof. Dr. Luiz Fernando Gomes Soares por toda a força que me concedeu para a realização deste trabalho. Sem o seu conhecimento e apoio não teria conseguido chegar ao fim desta jornada. A amizade do chefe também foi muito importante para motivar-me constantemente para realizar o melhor trabalho possível.

Aos amigos do Laboratório TeleMídia que fizeram parte da minha história no Mestrado, agradeço por estarem sempre ao meu lado, apoiando-me nos momentos difíceis. Quando nem eu mesmo acreditava no êxito, eles me incentivaram a prosseguir.

Aos funcionários da PUC-Rio, em especial os membros do Departamento de Informática, obrigado pela dedicação na resolução das minhas questões acadêmicas. A todos os professores do Departamento, agradeço pelas orientações e sabedoria das quais pude tirar muito proveito.

À CAPES e à PUC-Rio, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

Resumo

Junior, José Geraldo de Sousa Junior; Soares, Luiz F. G. **Uma Arquitetura para Aplicações Dinâmicas NCL Baseadas em Famílias de Documentos**. Rio de Janeiro, 2011. 72p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A apresentação de aplicações hipermídia dinâmicas pode ser vista em alguns casos como um processo de autoria recursivo, no qual aplicações são recriadas em tempo de apresentação quando modificações no conteúdo são desencadeadas por interações entre a máquina de apresentação e outras entidades (tais como usuários, objetos imperativos, aplicações externas etc.). Em alguns cenários de aplicações hipermídia dinâmicas, é possível identificar um padrão de composição hipermídia que se mantém mesmo após o documento ter sido recriado. Aplicações desse tipo são comuns, por exemplo, em um ambiente de TV Digital Interativa. A existência de um padrão durante a apresentação de uma aplicação para TV Digital Interativa permite estabelecer uma analogia entre os processos de recriar documentos dinamicamente e de especificar aplicações pelo método orientado a *templates*. Nesse último método, o autor especifica uma aplicação “preenchendo” lacunas deixadas por um *template* que representa o padrão de composição hipermídia da aplicação. Analogamente, na recriação dinâmica de documentos o módulo que processa atualizações no documento faz o papel de “preencher” as lacunas de *templates*. O objetivo principal do presente trabalho é definir uma arquitetura, inspirada nessa analogia, para estruturar aplicações NCL que podem ser refatoradas dinamicamente e que se mantêm em conformidade com seus respectivos *templates*. *Nested Context Language* (NCL) é a linguagem para autoria de aplicações do Sistema Brasileiro de TV Digital terrestre. Para validar a solução proposta, foram desenvolvidas uma aplicação que captura um cenário real de uso e uma ferramenta de autoria que permite especificar graficamente o preenchimento de *templates*.

Palavras-chave

NCL; Televisão Digital Interativa; Aplicações Dinâmicas; Templates; Autoria orientada a templates; Arquiteturas de software.

Abstract

Junior, José Geraldo de Sousa Junior; Soares, Luiz F. G. (Advisor). **An Architecture for Dynamic NCL Applications Based on Document Families**. Rio de Janeiro, 2011. 72p. MSc. Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The presentation of dynamic hypermedia applications may be seen as a recursive authoring process, in which applications are recreated during presentation time, whenever content changes are triggered by interactions between the presentation engine and other entities such as users, imperative objects, external applications, etc. In some scenarios of dynamic hypermedia applications, it is possible to identify a hypermedia composition pattern that remains consistent even after the document is recreated. This kind of applications is common, for instance, in an Interactive Digital Television environment. The presence of such a pattern throughout the presentation of an application for Interactive Digital Television allows the establishment of an analogy between recreating documents dynamically and authoring applications through a template-driven authoring method. Using the latter, the authoring process is conducted by “filling” gaps left by a template that represents the hypermedia composition pattern of an application. Analogously, in the dynamic document re-creation, the module that processes document updates fulfills the role of “filling” the templates gaps. The main goal of the present work is to define an architecture, inspired by this analogy, to structure NCL applications that can be dynamically refactored and that remain conform to their respective templates. Nested Context Language (NCL) is the language of Brazilian System of Digital Terrestrial Television applications. In order to validate the proposal, an application that captures a real scenario and an authoring tool for specify graphically document filling was developed.

Keywords

NCL; Interactive Digital Television; Dynamic applications; Templates; Template-driven authoring; Software architectures.

Sumário

1	Introdução	12
1.1.	Aplicações de TVD dinâmicas	14
1.2.	Motivação	17
1.3.	Objetivos	17
1.4.	Organização da dissertação	18
2	Geração Dinâmica de Conteúdo e Templates de Composição	19
2.1.	Conteúdos dinâmicos em sistemas hipermídia	19
2.2.	<i>Templates</i> de Composição Hipermídia	23
2.2.1.	Método de autoria orientado a <i>templates</i>	25
3	Trabalhos Relacionados	29
4	Arquitetura para aplicações NCL dinâmicas	33
4.1.	Exemplo de aplicação NCL que se recria dinamicamente	34
4.2.	Arquitetura da aplicação SAGGA1	35
4.2.1.	Visão de casos de uso	36
4.2.2.	Visão estrutural da aplicação	39
4.2.3.	Generalização da arquitetura	43
5	Implementação de Suporte à Geração Automática de Aplicações para o Ginga-NCL	44
5.1.	Especificação da aplicação SAGGA1	45
5.1.1.	Template e documento de preenchimento	46
5.1.2.	Recriando documentos SAGGA1 dinamicamente	49
5.1.3.	Ambiente de execução do gerador de conteúdo	51
5.2.	Ferramenta para autoria de documentos	53
5.2.1.	Diagrama de classes	54
5.2.2.	Funcionalidades	55

5.2.3. Etapas para a geração de um documento SAGGA	57
6 Conclusões e trabalhos futuros	59
7 Referências bibliográficas	62
ANEXO I – Cenários principais de casos de uso	65
ANEXO II – Template SAGGA1	66

Lista de figuras

Figura 2.1 - Visão estrutural de uma aplicação de TVD	20
Figura 2.2 - Visão estrutural de uma composição hipermídia	24
Figura 2.3 - Exemplo de <i>Template</i> de Composição	25
Figura 2.4 - Autoria de Documentos Hipermídia Orientada a <i>Templates</i> (Soares Neto, 2010)	27
Figura 2.5 - Método de autoria orientado a <i>templates</i> com gerador de conteúdo dinâmico	28
Figura 4.1 - Diagrama de casos de uso de SAGGA1	36
Figura 4.2 - Exemplo de documento NCL inicial de SAGGA1	37
Figura 4.3 - Visão estrutural de composição hipermídia SAGGA1	40
Figura 4.4 - Visão estrutural de SAGGA1 com gerador de conteúdo monolítico	41
Figura 4.5 - Visão estrutural completa de documentos SAGGA1	42
Figura 4.6 - Visão estrutural genérica para aplicações que se recriam dinamicamente	43
Figura 5.1 - Função handler do NCLua gerador de conte	45
Figura 5.2 - Trecho do <i>template</i> SAGGA1 para cardinalidade do vídeo principal	47
Figura 5.3 - Exemplo de comportamento recorrente	48
Figura 5.4 - Exemplo de documento de preenchimento de SAGGA1	49
Figura 5.5 - Aplicação SAGGA sendo exibida em momentos (a) e (b) seqüenciais	50
Figura 5.6 - Arquitetura para aplicações que se recriam dinamicamente com servidor <i>proxy</i>	52
Figura 5.7 - Diagrama de classes simplificado da ferramenta SAGGA	55
Figura 5.8 - Janela principal de SAGGA em dois momentos (a) e (b)	56
Figura 5.9 - Assistente para novo documento SAGGA	57
Figura 5.10 - Metadado em arquivo NCL informando o tipo de <i>template</i> de uma aplicação	58